DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

Assignment-10

Student Name: Mohit **UID: 22BCS14664**

Branch: CSE **Section: IOT_640(B)**

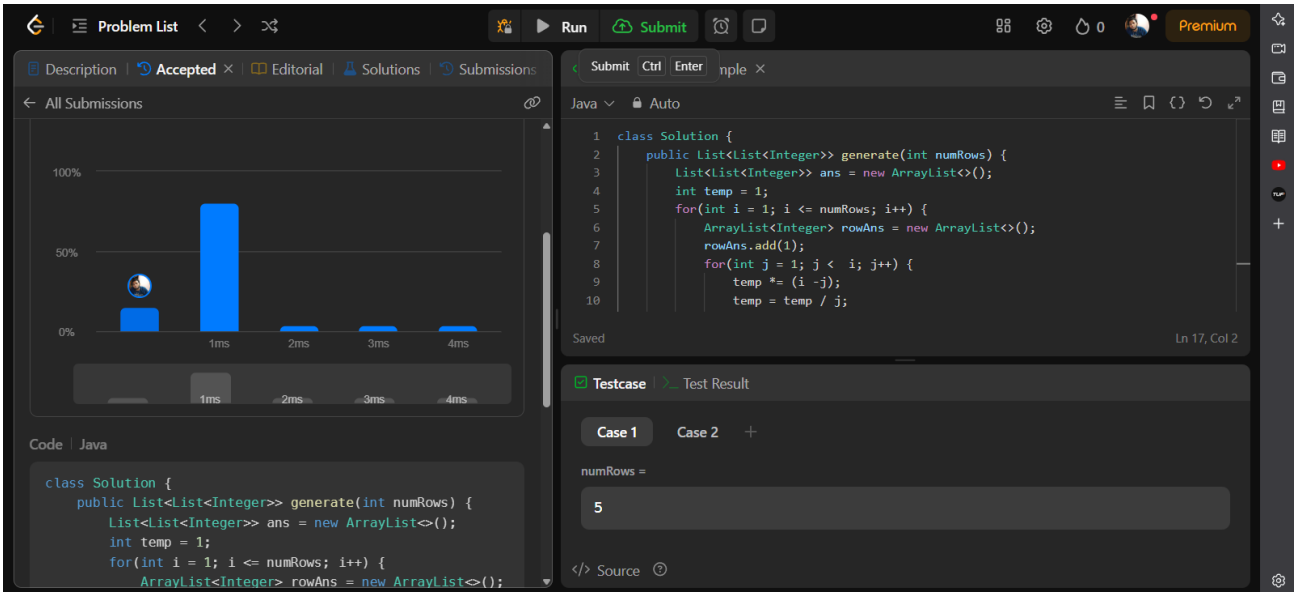
Semester: 6th **Date: 20/04/2025**

Subject: AP **Subject Code: 22CSP-351**

Problem 1:

```
class Solution {
    public List<List<Integer>> generate(int numRows) {
        List<List<Integer>> ans = new ArrayList<>();
        int temp = 1;
        for(int i = 1; i <= numRows; i++) {
            ArrayList<Integer> rowAns = new ArrayList<>();
            rowAns.add(1);
            for(int j = 1; j <  i; j++) {
                temp *= (i -j);
                temp = temp / j;
                rowAns.add(temp);
            }
            ans.add(rowAns);
        }
        return ans;
    }
}
```

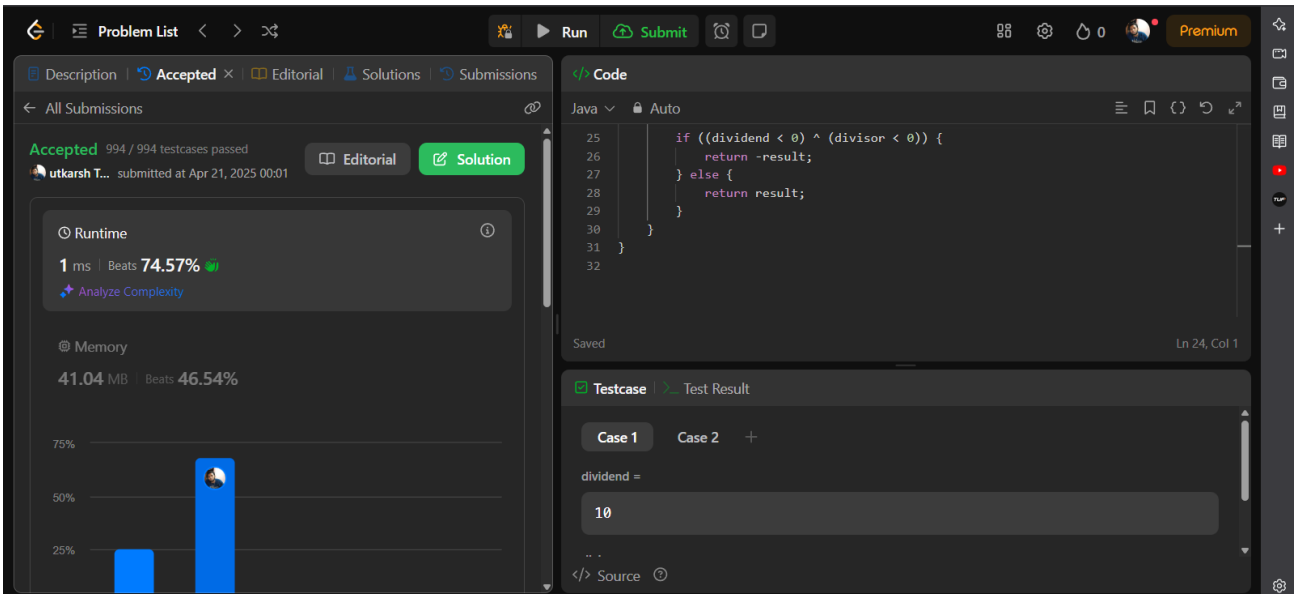
Screenshot:



Problem 2:

```
class Solution {
    public int divide(int dividend, int divisor) {

        if (dividend == Integer.MIN_VALUE && divisor == -1) {
            return Integer.MAX_VALUE;
        }

        long ldividend = Math.abs((long) dividend);
        long ldivisor = Math.abs((long) divisor);
        int result = 0;


        while (ldividend >= ldivisor) {
            long temp = ldivisor, multiple = 1;

            while (ldividend >= (temp << 1)) {
                temp <<= 1;
                multiple <<= 1;
            }

            ldividend -= temp;
            result += multiple;
        }

        if ((dividend < 0) ^ (divisor < 0)) {
            return -result;
        } else {
            return result;
        }
    }
}
```

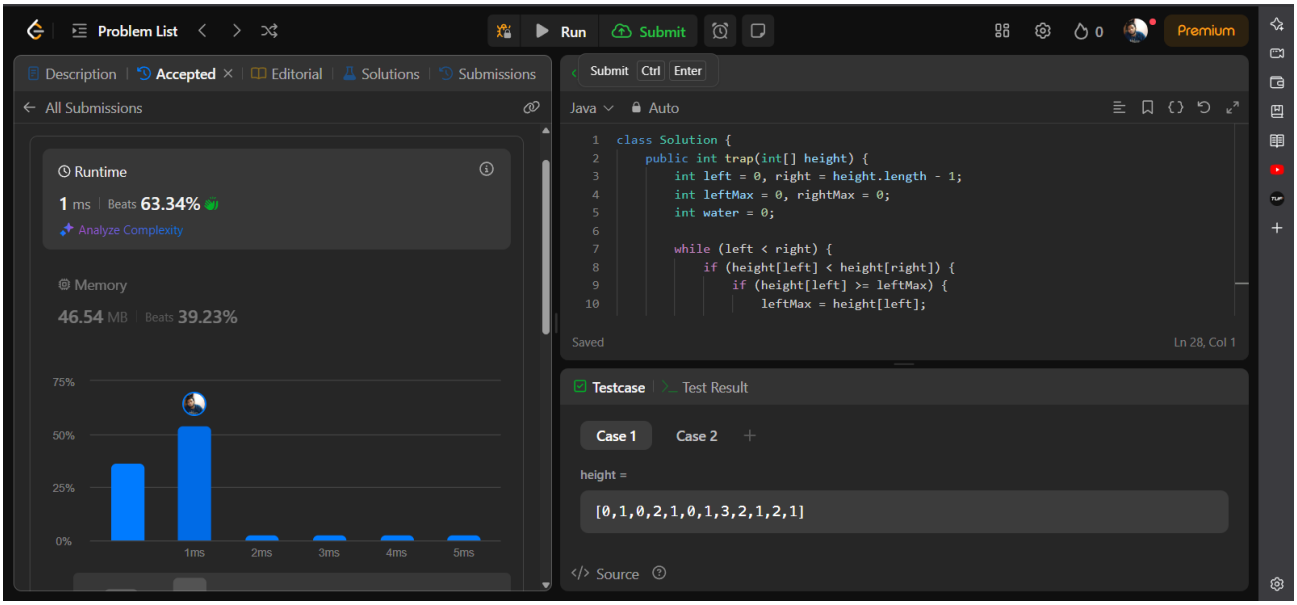
Screenshot:



Problem 3:

```
class Solution {
    public int trap(int[] height) {
        int left = 0, right = height.length - 1;
        int leftMax = 0, rightMax = 0;
        int water = 0;

        while (left < right) {
            if (height[left] < height[right]) {
                if (height[left] >= leftMax) {
                    leftMax = height[left];
                } else {
                    water += leftMax - height[left];
                }
                left++;
            } else {
                if (height[right] >= rightMax) {
                    rightMax = height[right];
                } else {
                    water += rightMax - height[right];
                }
                right--;
            }
        }

        return water;
    }
}
```

Screenshot:


Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Runtime
1 ms | Beats 63.34%
Analyze Complexity
Memory
46.54 MB | Beats 39.23%
75%
50%
25%
0%
1ms
2ms
3ms
4ms
5ms
Submit Ctrl Enter
Java
Auto
class Solution {
public int trap(int[] height) {
int left = 0, right = height.length - 1;
int leftMax = 0, rightMax = 0;
int water = 0;
while (left < right) {
if (height[left] < height[right]) {
if (height[left] >= leftMax) {
leftMax = height[left];
Saved
Ln 28, Col 1
Testcase
Test Result
Case 1
Case 2
height =
[0,1,0,2,1,0,1,3,2,1,2,1]
Source